9.5 Reactive-Transport Example

This one-dimensional example demonstrates the use of the reactive-transport module of FEHM. The input for the **rxn** and **trac** macros are those discussed in the example input for these macros in Section 6.2.41 and Section 6.2.50. The flow system is a one-dimensional flow path of 402 nodes (201 x 2) with rock properties and flow rates such that the mean fluid residence time in the path is 10,000 yr. Species *A* sorbs with a K_d of 0.1, which, for the rock properties chosen, is equivalent to a retardation factor of 5. The solute transport problem is run for 5,000 yr, or half of the mean residence time of the fluid. Therefore, in the absence of other reactions, species *A* would be expected to travel 0.5/5 = 0.1 of the length of the column.

When chemical reactions are included, the situation becomes more complex. Fig. 17 shows the concentration profiles at the end of the simulation for this example (called Run 1). Even though the first reaction is specified as kinetically controlled, the rate constants are large enough for the reaction to virtually reach equilibrium over the time period of the simulation ($k_r\tau = 50$, where τ is the time of the simulation). Thus, the concentrations of *A* and *B* essentially reach equilibrium; the equilibrium constant is the ratio of the rate constants, or 0.1 (there is 10 times as much *A* as *B* in solution). Of course, *A* is also present on the rock surface wherever concentrations are nonzero. Solute *C* travels with *A* and *B* and is in equilibrium with *B* in solution; its concentration is 0.2 times that of *B* everywhere because of the equilibrium constant chosen. The entire suite of solutes has moved roughly 10% of the way down the column, as discussed above. With chemical

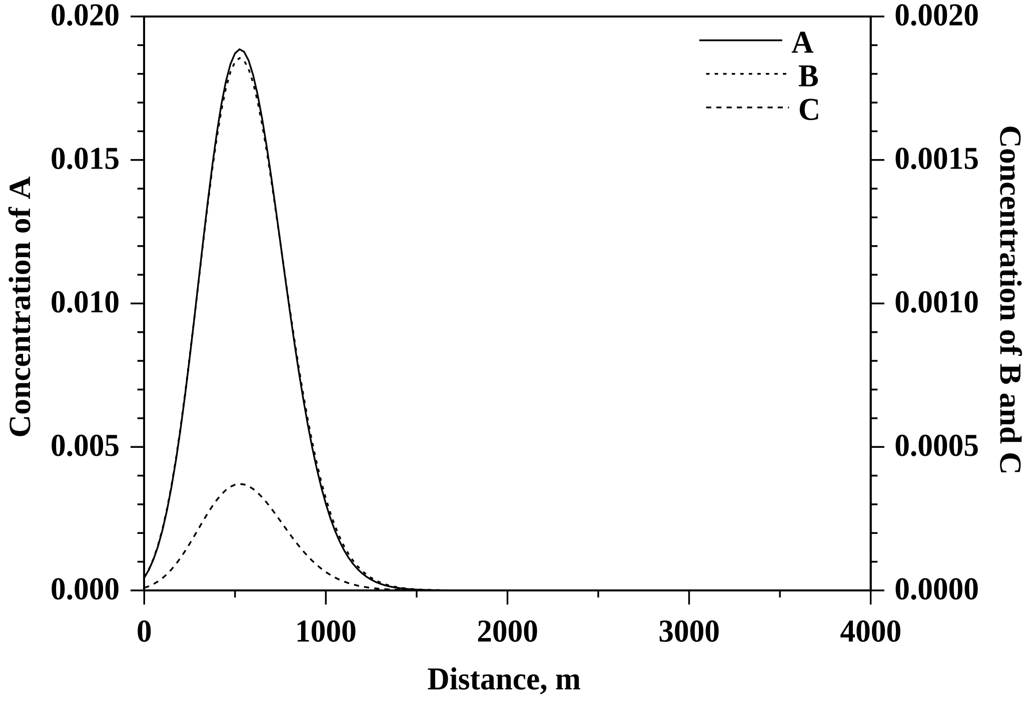

Figure 16. Concentration profiles for Run 1.

reactions, the rate of movement of the solutes can be faster than that of a non-reacting species that only sorbs, because solutes *B* and *C* do not sorb. However, for this example, solute *A* constitutes about 90% of the aqueous portion of the contaminant. Kinetics and equilibrium parameters that favor the formation of *B* and *C* would in turn increase the rate of movement of the contaminants.

A common application is radionuclide migration, for which we must include radioactive decay of each solute. It may also be of interest to track the movement of the daughter product. In Run 2, we assume that all species of Run 1 are species of the same radionuclide, each of which undergoes decay to a long-lived, mobile radionuclide that does not sorb. The input file is changed to include a fourth solute in the **trac** macro that is identical in input to the second and third solutes. The changes to the **rxn** macro are more complex and thus are shown in Fig. 17.

```
rxn
group
2
1            1            1            0
0            0            0            1
5            0
kinetic      3.1688e-11   0.           3.1688e-10   0.
equilibrium  1            0.2          0.           1.e-2     1.e-3     1.e-10
kinetic      3.1688e-12   0.           0.           0.
kinetic      3.1688e-12   0.           0.           0.
kinetic      3.1688e-12   0.           0.           0.
1           -1            0            0
0            1           -1            0
1            0            0           -1
0            1            0           -1
0            0            1           -1
1           -1            0            0
0            1           -1            0
1            0            0           -1
0            1            0           -1
0            0            1           -1
1            1            1            1
1            1            1            1
1            1            1            1
1            1            1            1
1            1            1            1
0            0            0            0
0            0            0            0
1            1            1            1
0            0            0            0
0            0            0            0
```

Figure 17. FEHM macro rxn in the input file for reactive-transport example.

Radioactive decay is handled by adding three reactions, one each for *A*, *B*, and *C* reacting to form D. The kinetics of each of these reactions are identical, reflecting an irreversible, first-order decay to form the daughter product. Solute *A* sorbs to the rock surface; therefore, we must specify that the decay reaction takes place for both the aqueous and solid portions of the solute. This is done by setting FL_MULT and SB_MULT to 1 for the third reaction.

Figure 18 shows the results of this example. A significant amount of daughter product D has been produced, and it is not forced to travel at the reduced velocity of sorbing solute *A* because it is connected to the other solutes only through the radioactive decay source term. The original solutes behave as they did in Run 1, except that their concentrations are somewhat reduced due to radioactive decay.

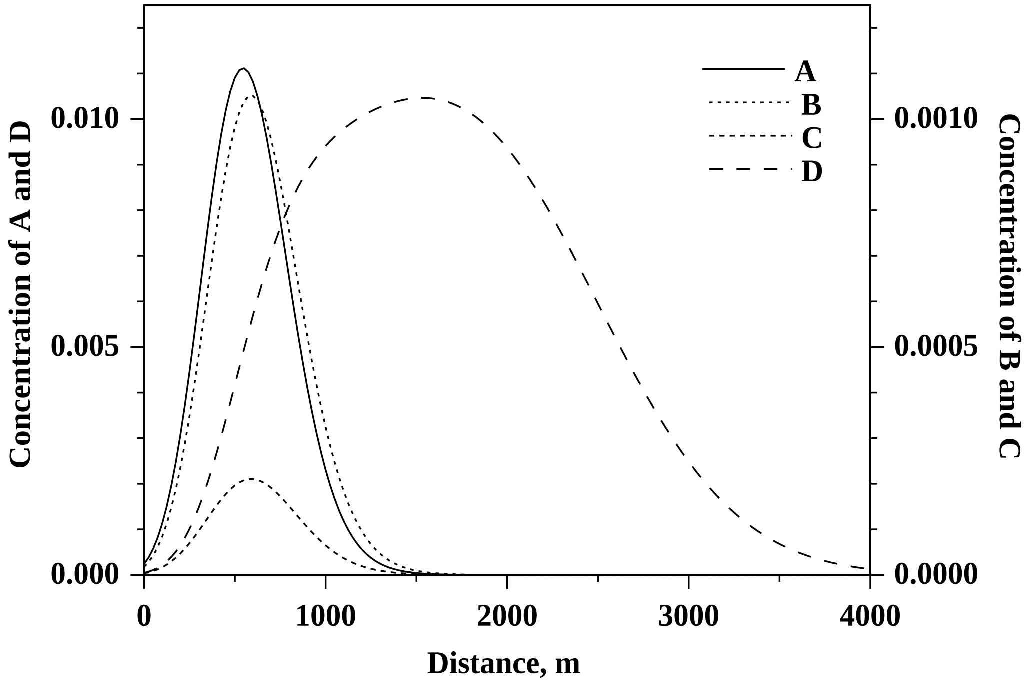

Figure 18. Concentration profiles for Run 2.

Considerations of Numerical Efficiency. The decision of how to group the solutes represents a trade-off among the robustness required for a given reactive transport system, memory requirements, and computational speed. We use these example runs to illustrate some of the considerations. In the discussions below, when we speak of the coupling of solutes, we refer to the method of grouping the solutes into systems of equations that are solved simultaneously. Regardless of the grouping of and order in which the solute concentrations are solved, the code requires that the full system of interacting solutes reach convergence at every time step. The overall solution is therefore “fully coupled,” regardless of the details of the solution procedure.

In Run 1, we solved first for solute *A* alone, after which solutes *B* and *C* were coupled. Solutes that are coupled only through a kinetic reaction need not be solved simultaneously as long as the kinetics are not too rapid. When systems are solved as more than one group, the time required to complete one outer iteration is shorter, but more outer iterations will be required (only one outer iteration is required if all solutes are coupled into a single group). A rule of thumb is that as long as only a few outer iterations are required, solving the problem in several groups will be competitive with a more fully coupled solution. For example, in

Run 1, two or three outer iterations were typically required. Coupling all three solutes reduces the number of outer iterations to 1, but the computational time was virtually identical in this example problem. Note that solutes coupled through equilibrium reactions must always be solved simultaneously because these reactions are specified, within the code, with very rapid kinetics to approximate equilibrium behavior. An added benefit of solving the system as several groups is that the memory requirements are lower. This factor was not a consideration for this example, which solved a system of only 402 nodes.

In Run 2, notice that the fourth solute was solved alone subsequent to a group that coupled the first three. There is no benefit to the convergence of the system of equations from simultaneously solving solute D with the other three because it is formed only from irreversible reactions (radioactive decay) involving solutes *A*, *B*, and *C*. This fact means that there is no “feed-back” from the concentration of D onto the other concentrations. The decay reactions provide the source term for solute D, but the concentration of D does not impact the solution of the other solutes; the coupling is one-way. The only potential benefit is that when all solutes are coupled, the system is automatically solved in a single outer iteration, whereas the code cannot assume overall convergence and must perform a second outer iteration when D is decoupled from the other three solutes. For this problem, the benefit of fully coupling the solution is almost exactly counterbalanced by the additional work of finding a four degree-of-freedom solution (versus three followed by one), so that the two solutions take comparable times to finish.